Application-Specific Soft Processor Optimization through Instruction-Level Frequency Scaling

Sorin Dan Cotofana

Computer Engineering Laboratory
Faculty of Electrical Engineering, Mathematics and Computer Science
Delft University of Technology (TU Delft), The Netherlands

TUDelft

1

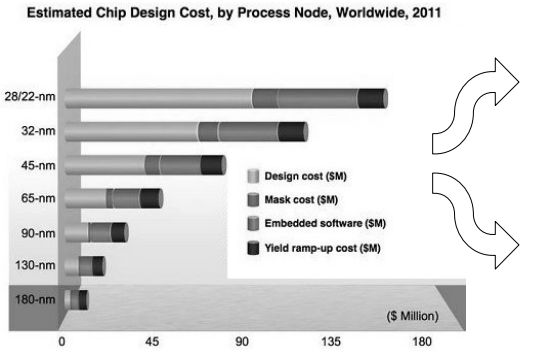
Overview

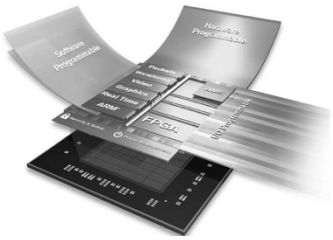
- Motivation
- FPGA Soft Processors as Accelerators
- Application-Specific Soft Processors
  - ISA Customization and Optimization
  - Instruction-Level Frequency Scaling
- Conclusions

TUDelft

2

Motivation

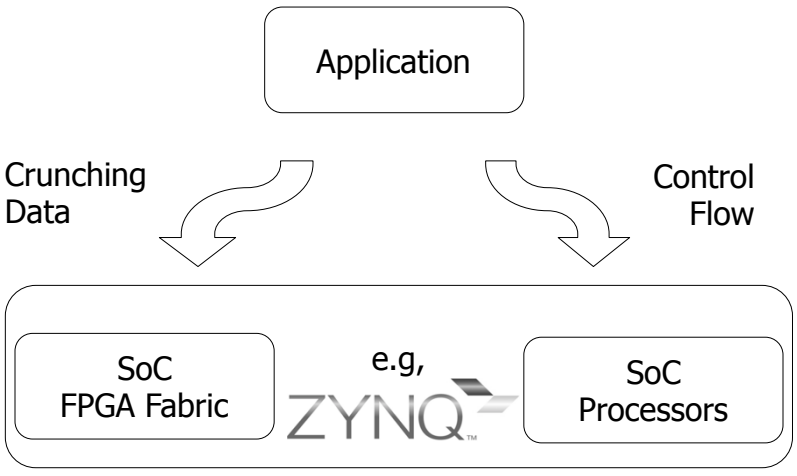




TUDelft

3

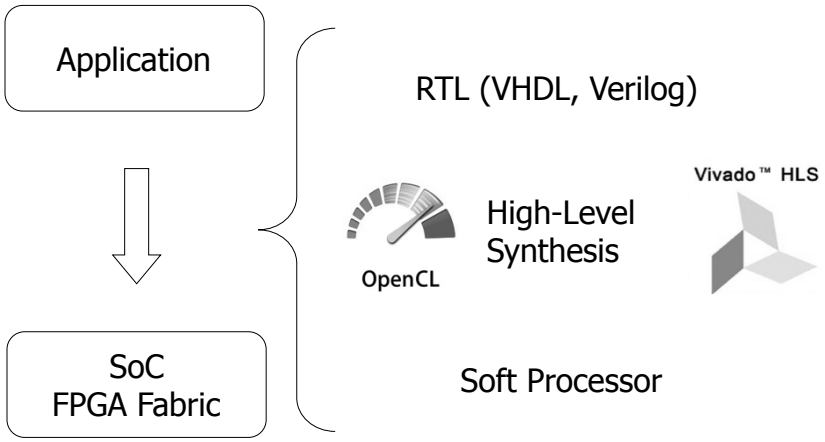
Leveraging FPGA MPSoCs



TUDelft

4

Leveraging FPGA MPSoCs



TUDelft

5

Soft Processors as Accelerators

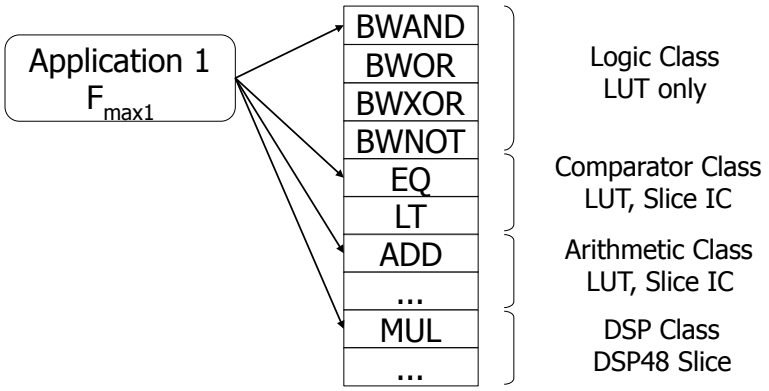
Microblaze, Nios
- Intended for control in uP-less FPGAs
- Limited ISA customization (FP/DIV on/off)
- Limited Optimization (Pipeline Depth)

Delft ρ-Vex, UPB Opincaa
- Parallel (VLIW, SIMD respectively)
- Intended for acceleration
- High ISA customization
- Advanced Optimization

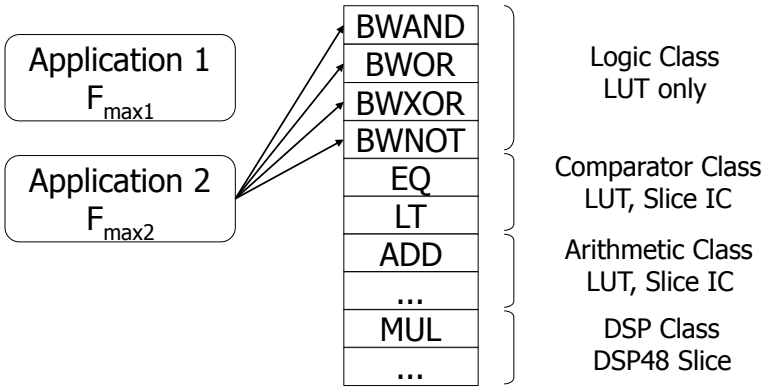
TUDelft

6

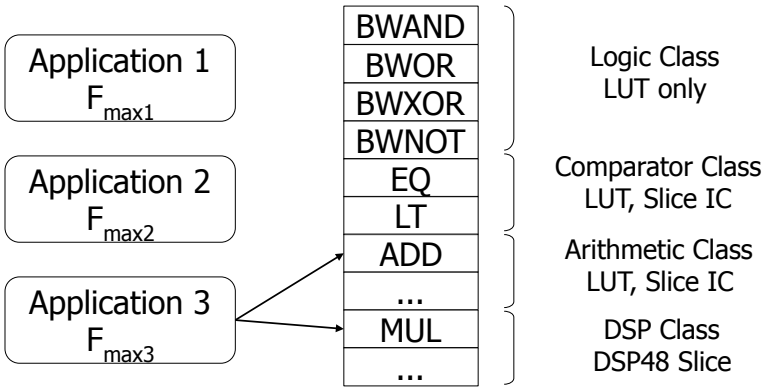
ISA Customization and Optimization



ISA Customization and Optimization



ISA Customization and Optimization



ISA Customization and Optimization

Unused Instruction Elimination

- Reduces congestion from useless logic
- Enables tighter placement of useful logic
- Instructions eliminated by class
- Code compilation/optimization should be aware of possible ISA customizations
  - E.g., sum of absolute differences may be implemented with add/sub/compare, logic/add, or add/lookup

ISA Customization and Optimization

E.g., 128 SIMD ISA Customization on Zynq 7020
- Full ISA, F_{max} = ~80 MHz (highly congested)
- Shift instruction removed, F_{max} = ~125 MHz
- Add/Mul only, F_{max} = ~200 MHz

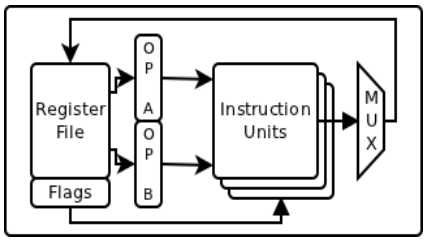
SIMD Lane Width Customization
- Full ISA, SIMD lane width 128 (congestion)
- Add/Mul only, SIMD lane width 220 (DSP48)
- No Mul/Logic/Shift, SIMD lane width 276 (Block RAM)

TUDelft

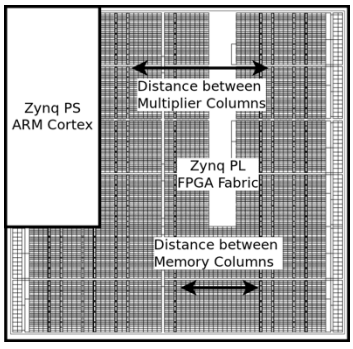
11

Instruction-Level Frequency Scaling

- Instructions occur with varying rates
- Longest path delay dictates frequency
- No correlation between logic delay of instruction and occurrence rate



E.g. 128 SIMD on Zynq,
MUL incurs high routing delay

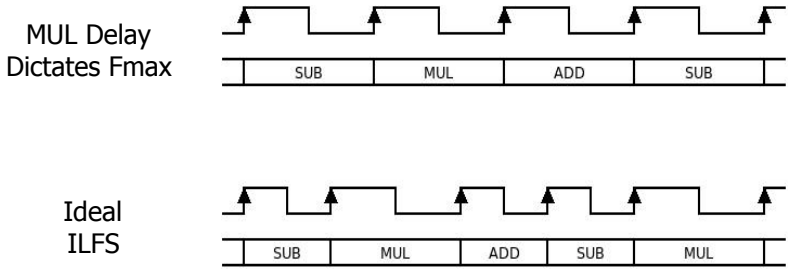


TUDelft

12

Instruction-Level Frequency Scaling

ILFS – decouple longest delay and top frequency
- E.g., 128 SIMD on Zynq 7020, F_{max} 125 MHz
- MUL paths ignored, F_{max} ~160 MHz

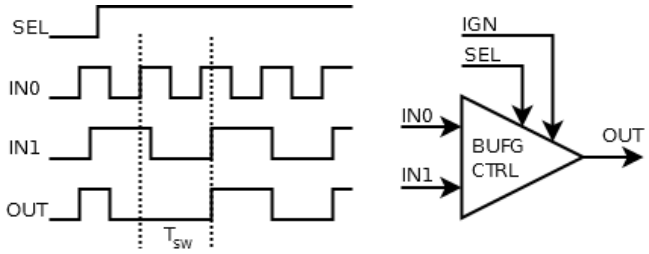


TUDelft

13

Instruction-Level Frequency Scaling

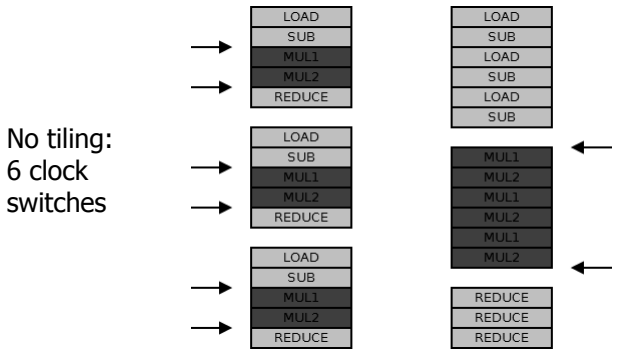
Zynq Implementation
- Uses BUFGCTRL clock multiplexer
- Incurs multiplexing latency at every clock frequency switch



TUDelft

14

Instruction-Level Frequency Scaling



No tiling: 6 clock switches

With tiling: 2 clock switches

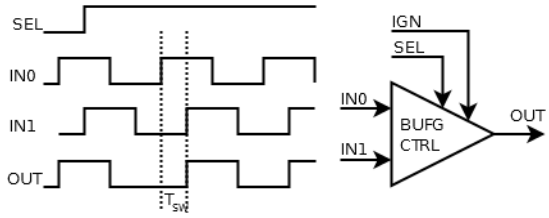
Tiling limited by number of registers and data dependency !

TUDelft

Instruction-Level Frequency Scaling

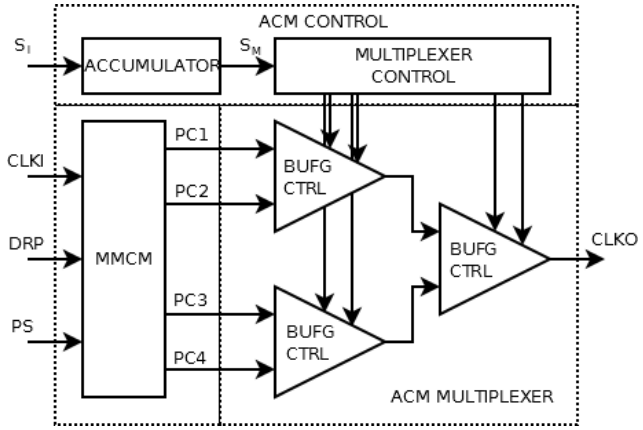
BUFGCTRL predictable when clocks have same F

- T_{SW} equals phase offset (source to destination)
- Phase offset known, T_{SW} is controlled stretch
- Continuous switch == continuous stretch
- Zero-latency T extension by T_{SW}



TUDelft

Instruction-Level Frequency Scaling



Instruction-Level Frequency Scaling

Automatic ILFS Configuration

- Profile application, determine instruction occurrence rates
- Apply separate timing constraints to paths pertaining to each instruction
- Tighten constraints for common instructions, loosen for rare instructions, synthesize
- Repeat while timing constraints met

Conclusion

- Soft processors an alternative to HLS or RTL
- ISA Customization enables application-specific soft processors, higher F_{max}, wider SIMD
- Instruction-Level Frequency Scaling possible with existing FPGAs, enables higher F_{max} even under difficult routing

Open Issues
- Customization-aware compiling
- ILFS per-instruction optimization difficult with current FPGA synthesis tools

Acknowledgement

Electronic Devices, Circuits and Architectures Department, Politehnica University of Bucharest (http://www.dcae.pub.ro/en/)

Opincaa Team:
- Radu Hobincu
- Călin Bîră
- Vlad Popescu
- Alexandru Gheolbănoiu
- Lucian Petrică (lucian.petrica@upb.ro)